**Fourier Series**

The **Fourier series** of a $2L$periodic function $f(x)$is given by

$$\frac {a_0} 2 +\sum _{k=1}^\infty \left( a_k \cos \frac{k\pi x}L
  + b_k \sin \frac{k \pi x}L\right)$$

where the **Fourier coefficients** $a_k,\ k=0,1,2,\dots,$and $b_k,\ k=1,2,3, \dots,$are given by

$$a_k = \frac 1 L \int _{\mbox{--} L}^L f(x) \cos \frac {k\pi x}L\,dx$$

and

$$b_k = \frac 1 L \int _{\mbox{--} L}^L f(x) \sin \frac {k\pi x}L\,dx\,.$$

The nth partial sum of the Fourier series is

$$s_n(x) = \frac {a_0} 2 +\sum _{k=1}^n \left( a_k \cos \frac{k\pi x}L
  + b_k \sin \frac{k\pi x}L\right).$$

You can use the following commands to calculate the nth partial sum of the Fourier series of the expression f on the interval [-L,L]

syms x k L n

The next command tells MATLAB that k is an integer. That will allow **simple** and **simplify** to evaluate $\sin(k\pi)$and $\cos(k\pi)$for a symbolic integer k.

evalin(symengine,'assume(k,Type::Integer)');

The kth Fourier cosine coefficient $a_k$of f is given by the command:

a = @(f,x,k,L) int(f\*cos(k\*pi\*x/L)/L,x,-L,L);

The kth Fourier sine coefficient $b_k$is given by the command:

b = @(f,x,k,L) int(f\*sin(k\*pi\*x/L)/L,x,-L,L);

The nth partial sum is given by

fs = @(f,x,n,L) a(f,x,0,L)/2 + ...

symsum(a(f,x,k,L)\*cos(k\*pi\*x/L) + b(f,x,k,L)\*sin(k\*pi\*x/L),k,1,n);

For example, I can calculate the Fourier series of f(x) = |x| on the interval [-1,1].

f = abs(x)

f =

abs(x)

The 10th partial sum is

pretty(fs(f,x,10,1))

4 cos(3 pi x) 4 cos(5 pi x) 4 cos(7 pi x) 4 cos(9 pi x) 4 cos(pi x)

1/2 - ------------- - ------------- - ------------- - ------------- - -----------

2 2 2 2 2

9 pi 25 pi 49 pi 81 pi pi

We can also have MATLAB calculuate the general Fourier coefficients. To do this and get MATLAB to simplify the results, we can use **simple**. The following command gives the kth Fourier cosine coefficient, suppressing the results of all of the steps of **simple** except for the simplest.

[A,how]=simple(a(f,x,k,1))

A =

-(4\*sin((pi\*k)/2)^2)/(pi^2\*k^2)

how =

simplify

If I don't want to see how **simple** found the answer, I can suppress the output, then just display the simplified answer. The following command does that for the kth Fourier sine coefficient.

[B,how]=simple(b(f,x,k,1)); B

B =

0

Here are the plots of the partial sums for n=2,5,10. The plot also shows the function f.

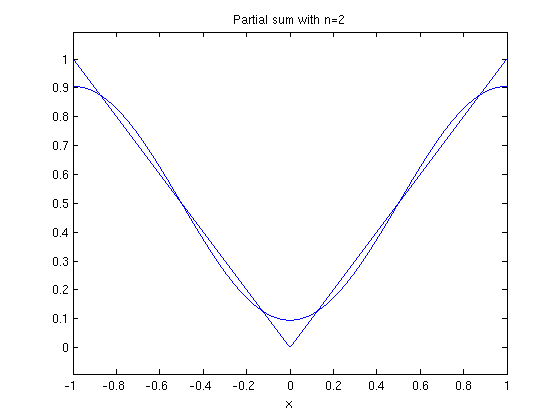
ezplot(fs(f,x,2,1),-1,1)

hold on

ezplot(f,-1,1)

hold off

title('Partial sum with n=2')



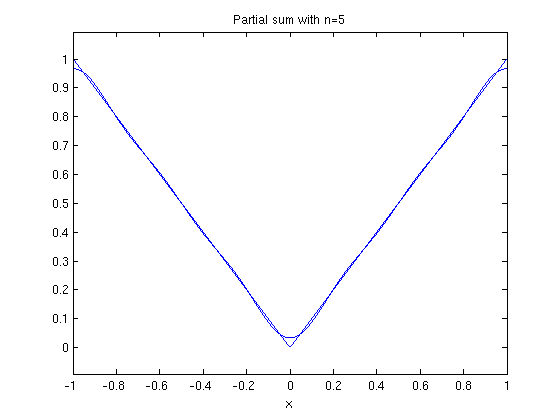
ezplot(fs(f,x,5,1),-1,1)

hold on

ezplot(f,-1,1)

hold off

title('Partial sum with n=5')



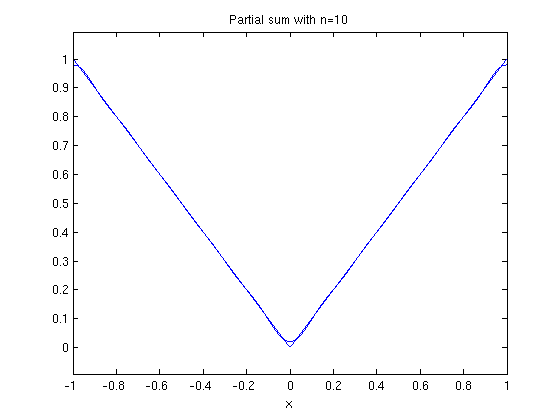
ezplot(fs(f,x,10,1),-1,1)

hold on

ezplot(f,-1,1)

hold off

title('Partial sum with n=10')



Now I do it with the function f(x) = x on [-1,1].

f = x

f =

x

The Fourier cosine coefficients are

[A,how]=simple(a(f,x,k,1)); A

A =

0

The Fourier sine coefficients are

[B,how]=simple(b(f,x,k,1)); B

B =

-(2\*(-1)^k)/(pi\*k)

The 10th partial sum is

pretty(fs(f,x,10,1))

2 sin(pi x) sin(2 pi x) 2 sin(3 pi x) sin(4 pi x) 2 sin(5 pi x) sin(6 pi x)

----------- - ----------- + ------------- - ----------- + ------------- - ----------- +

pi pi 3 pi 2 pi 5 pi 3 pi

2 sin(7 pi x) sin(8 pi x) 2 sin(9 pi x) sin(10 pi x)

------------- - ----------- + ------------- - ------------

7 pi 4 pi 9 pi 5 pi

Here are plots of the partial sums for n=2,5,10,20,50.

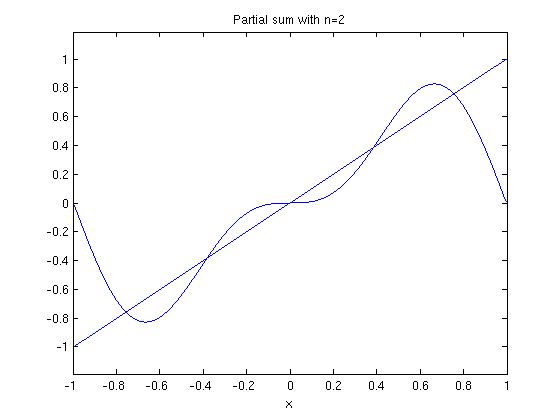
ezplot(fs(f,x,2,1),-1,1)

hold on

ezplot(f,-1,1)

hold off

title('Partial sum with n=2')



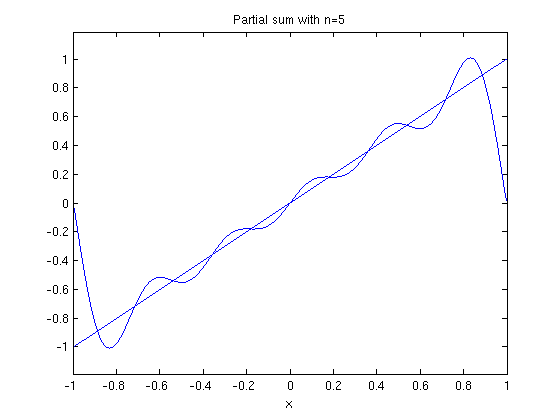
ezplot(fs(f,x,5,1),-1,1)

hold on

ezplot(f,-1,1)

hold off

title('Partial sum with n=5')



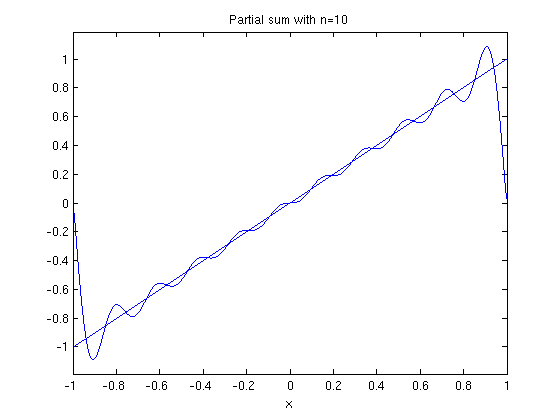
ezplot(fs(f,x,10,1),-1,1)

hold on

ezplot(f,-1,1)

hold off

title('Partial sum with n=10')



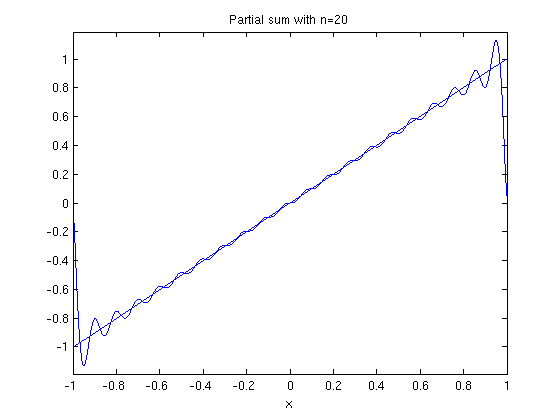
ezplot(fs(f,x,20,1),-1,1)

hold on

ezplot(f,-1,1)

hold off

title('Partial sum with n=20')



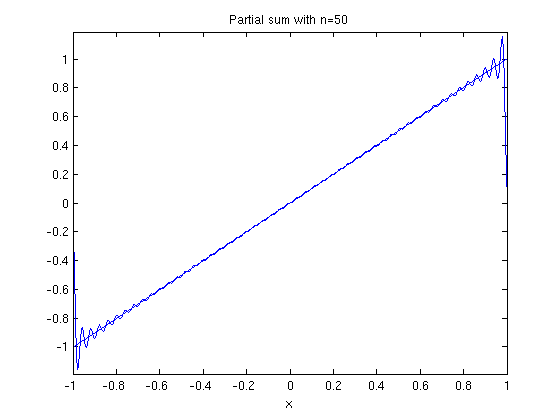
ezplot(fs(f,x,50,1),-1,1)

hold on

ezplot(f,-1,1)

hold off

title('Partial sum with n=50')



If you zoom in here, you notice that the graph of the partial sum of the Fourier series looks very jagged. That is because **ezplot** does not plot enough points compared to the number of oscillations in the functions in the partial sum. We can fix this problem using plot. This also allows us to use a different colors for the plot of the original function and the plot of the partial sum. To use plot, we need to turn the partial sum into an inline vectorized function and specify the points where it will be evaluated.

g = inline(vectorize(fs(f,x,50,1)));

X = -1:.001:1;

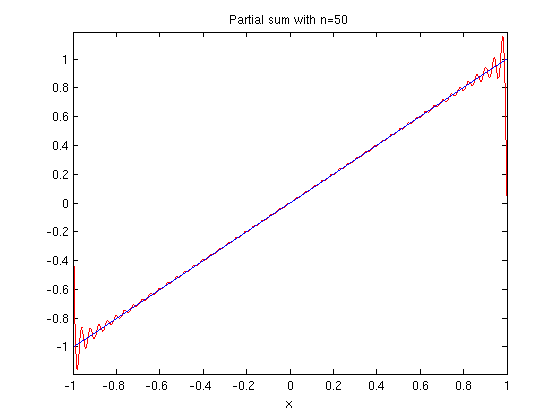
plot(X,g(X),'r')

hold on

ezplot(f,-1,1)

hold off

title('Partial sum with n=50')



g = inline(vectorize(fs(f,x,100,1)));

X = -1:.0001:1;

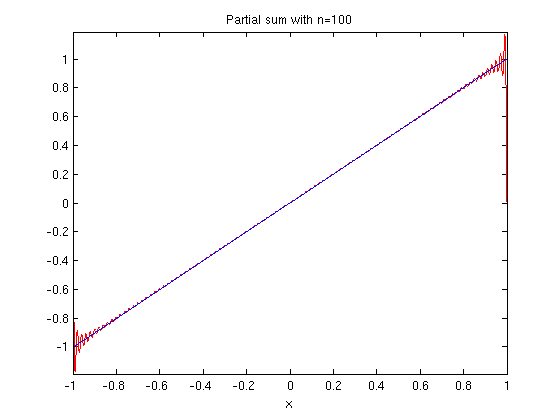
plot(X,g(X),'r')

hold on

ezplot(f,-1,1)

hold off

title('Partial sum with n=100')



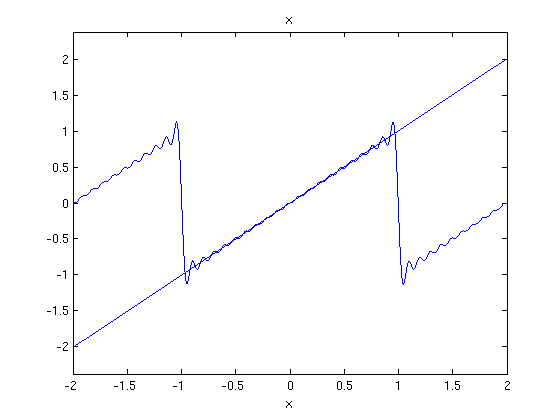
Notice that no matter how many terms we take, the partial sum always overshoots the function x at the end points of the interval. If we do our two plots on the interval [-2,2], we see that outside [-1,1], the partial sum doesn't have much to do with the function.

ezplot(fs(f,x,20,1),-2,2)

hold on

ezplot(f,-2,2)

hold off



A Fourier series on [-L,L] is 2L periodic, and so are all its partial sums. So, what we are really doing when we compute the Fourier series of a function f on the interval [-L,L] is computing the Fourier series of the 2L periodic extension of f. To do that in MATLAB, we have to make use of the unit step function u(x), which is 0 if $x < 0$and 1 if $x \ge 0$. It is known as the Heaviside function, and the MATLAB command for it is **heaviside**. In MATLAB, heaviside(0)=1/2. The function h(x) = u(x-a)u(b-x) is 1 on the interval [a,b] and 0 outside the interval.

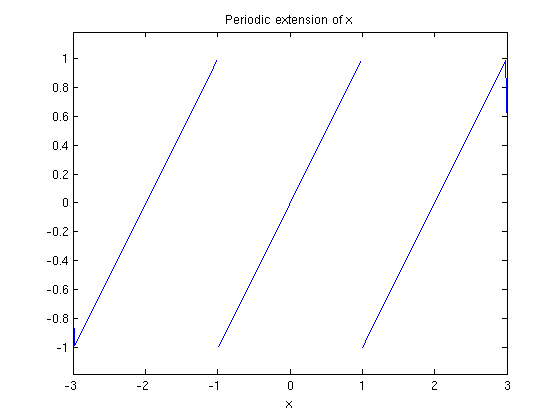
f = heaviside(x+3)\*heaviside(-1-x)\*(x+2) + heaviside(x+1)\*heaviside(1-x)\*x ...

+ heaviside(x-1)\*heaviside(3-x)\*(x-2);

extends f(x) = x to be periodic on [-3,3], with period 2. To check that we've extended it correctly, we plot it.

ezplot(f,-3,3)

title('Periodic extension of x')



Here is a plot of the function and its Fourier series.

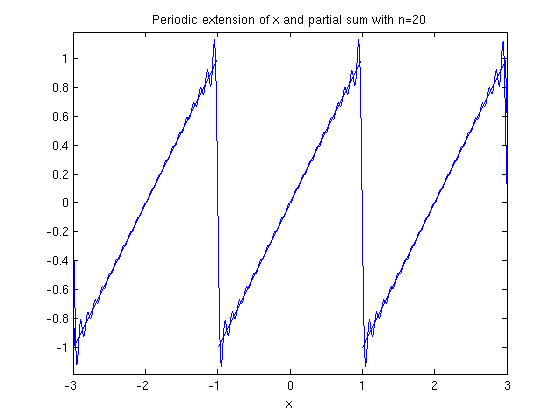
ezplot(fs(x,x,20,1),-3,3)

hold on

ezplot(f,-3,3)

hold off

title('Periodic extension of x and partial sum with n=20')



It is somewhat clearer if the plots aren't in the same color, so I'll use **plot** for the partial sum.

X = -3:0.001:3;

g = inline(vectorize(fs(f,x,20,1)));

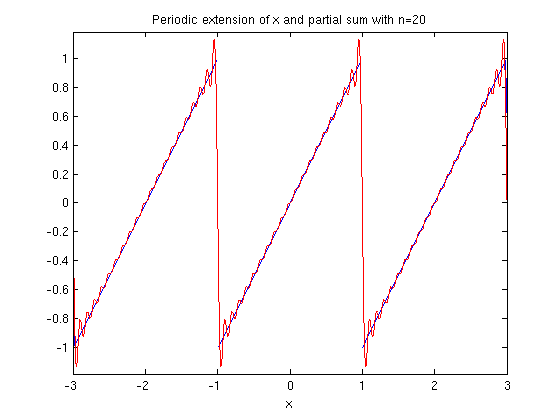
ezplot(f,-3,3)

hold on

plot(X,g(X),'r')

hold off

title('Periodic extension of x and partial sum with n=20')



The overshoots are at the discontinuities of the 2 periodic extension of f. This is called the **Gibbs Phenomenon.**

Published with MATLAB® 7.13